Il nuovo Amiga 500 Plus

TUTOR

- Amos
- Amiga PostScript
- Visual Basic

ASSEMBLY/C64

- Bolle pazze

PROVE

- Olidata MDS 286/20
- Aashima MagicBook 386 SX
- Dell 320 N
- Citizen PN 48
- Microsoft Ballpoint

SOFTWARE

- Nuovo compilatore Amos
- Calcolatrice scientifica in Clipper
- Animazioni 3D
- Windows 3.1
- Pascal per Amiga

Anno XI - N. 90 - 25 dicembre 1991 - 25 gennaio 1992 - Sped. Abb. Post. Gr III/70 - CR - Distr.: Parrini

Systems

Bolle pazze per il C/64

< di Filippo Bruno >

< E' l'ultimo articolo sul C/64? Forse no... >

Dopo aver parlato, nei numeri precedenti, di caratteri rimbalzanti e di effetti sonori in interrupt, è quasi d'obbligo affrontare il tema del **movimento complesso di sprite** (o mobs che dir si voglia) in interrupt.

Chi di noi (fra quelli che si sono spinti oltre lo sprite a forma di mongolfiera presente sul manuale del C/64) infatti, non ha mai assistito con invidia a giochi in cui frotte di sprite si muovono seguendo una sinusoide o una figura geometrica?

Per non parlare poi dei giochi spaziali, dove si vedono nugoli di stelle che ci vengono incontro a raggiera da un punto virtualmente all'infinito.

Stavolta simuliamo, quindi, un movimento che porti otto corpi (gli sprite) da una profondità "infinita" a quella del video (con conseguente aumento delle dimensioni) senza contare che tale moto dovrà essere a raggiera e *nel contempo* rotatorio, divenendo così elicoidale. Il tutto in interrupt: e non è poco.

Problemi

Passando dalle parole ai fatti, subito andiamo incontro a numerosi problemi. Il più pressante è senza dubbio la velocità; è impensabile calcolare, istante per istante, le coordinate di 8 sprite perché ciò andrebbe a discapito dell'effetto estetico finale, anche lavorando in lm.

In più occorrerebbe studiare un algoritmo talmente complesso che occuperebbe metà rivista.

Occorre allora *stabilire a priori quale dovrà essere il percorso degli sprite* e memorizzare le loro coordinate, passo dopo passo, in specifiche tabelle (necessariamente un po' lunghe) allocate da qualche parte in memoria, possibilmente dove non diano fastidio.

Proprio per questo motivo non vi deve scoraggiare, in partenza, la quantità apparentemente elevata di linee data presenti nei 5 listati basic.

Innanzi tutto perché alcune sono facoltative (a patto di limitarsi all'*effetto stelle* e di non provare l'*effetto bolle* di gran lunga più entusiasmante) e poi perché, una volta capito il trucco, potrete creare voi stessi nuovi effetti senza ricorrere ad esse.

Il programma lm completo, *bolle pazze*, occupa l'intera area da **$C000** a **$CFFF** (da **49152** a **53247**) e contiene tutte le mappe e le tabelle necessarie; la routine vera e propria è però allocata da $C000 a $C367.

Non potendo inserire anche i dati dei **26 sprite** delle bolle nella stessa zona, occupano 1664 byte, sono stati allocati da $0800 (2048) a $0F7F (3711).

I più perspicaci avranno intuito che si dovranno **modificare i puntatori di inizio basic** (che inizia, appena acceso il computer, proprio a partire da $0800) pena la sicura sostituzione degli sprite con sporcizia varia (in realtà i token del basic, le variabili, etc.).

Le tabelle necessarie sono 10 in totale e servono per memorizzare:

1) le coordinate **x** e **y** del moto rotatorio;
2) le coor. **x** e **y** del moto a raggiera;
3) la figura dello sprite;
4) l'eventuale superamento del valore 255 per le coordinate **x** degli sprite.

La maggior parte delle tabelle sono presenti nei listati basic in forma compattata: in pratica il listato si occuperà di scrivere un programma in lm che, una volta lanciato, scriverà la tabella; ovviamente tale programma risulta essere più corto della tabella stessa, altrimenti non ci sarebbe stato alcun vantaggio.

Come operare

Passiamo ora alla spiegazione della procedura da compiersi per coloro che desiderassero accedere direttamente al programma finale pronto per l'uso

(magari senza capire come e perché funziona) rimandando la spiegazione dettagliata.

Dovrete innanzitutto copiare i cinque listati e salvarli su disco, **senza** lanciarli. Vi assicurerete, **dopo il lancio**, che non vi siano errori di sorta; se vi sono, correggeteli e risalvate il tutto. Digitate...

```
poke 44, 12: poke 3072,0
new
load "listato 1", 8
```

...e seguite le istruzioni che verranno visualizzate.

Caricate e lanciate, poi, il listato n. 2 e seguite anche stavolta le istruzioni.

A questo punto avete presente, su disco, un programma di 7 blocchi, di nome ***Mobs Bolle***, contenente gli sprite delle bolle a partire da $0800.

Spegnete e riaccendete, digitando subito dopo...

```
poke 44, 16: poke 4096, 0
new
load "listato 3", 8
```

...lanciatelo e seguite le istruzioni, badando bene che quando queste ordinano di **resettare** *non* bisogna spegnere e riaccendere il computer, ma solo **digitare** *sys 64738* o premere l'apposito pulsante di reset (per chi ce l'ha).

Caricate e lanciate quindi il listato **n. 4** e seguite le istruzioni tra cui la scelta dell'effetto (stelle o bolle) e la velocità del medesimo.

Infine caricate il listato **5** e lanciatelo: avrete finalmente ottenuto su disco il **programma completo** di nome ***Bolle Pazze***.

Ora spegnete e riaccendete, quindi digitate su un unico rigo...

```
poke 44, 14: poke 43, 129:
poke 3712, 0: new
```

...per spostare l'inizio del basic a partire da $0E80. Con il suffisso ,8,1 caricate il file contenente gli sprite (chiamato *Mobs Bolle*) e quello contenente la routine (*Bolle Pazze*), infine impartire un altro **New**.

E' finalmente giunto il momento di lanciare la routine, con **sys 49152**, che opera in interrupt; ciò significa che potete editare un programma, listarlo, modificarlo mentre sullo schermo viaggiano le bolle.

Non vi consigliamo di usare il drive durante la visualizzazione delle bolle, dal momento che, come è noto, la presenza contemporanea di 8 sprite genera un malfunzionamento nell'uso delle routine di I/O. Per interrompere la visualizzazione, comunque, è sufficiente un banale *Run / Stop + Restore*.

La scelta dell'*effetto stelle* ha uno scopo puramente dimostrativo, pertanto tutto risulta ridotto all'essenziale, consentendovi di **saltare la trascrizione dei listati 1** e **2** che riguardano solo le "figure" delle bolle.

Il funzionamento della routine che si occupa del moto degli sprite è commentato abbondantemente nel disassemblato, per cui spiegheremo solamente l'essenziale.

La routine Im

La routine consta essenzialmente di 3 parti: nella prima vengono inizializzati alcuni registri (come quelli del colore) relativi agli sprite e viene deviato il puntatore dell'interrupt all'inizio della seconda parte.

Quest'ultima si occupa, invece, di regolare la velocità di esecuzione: attivando due volte, per ogni "scatto" raster, la routine **move**, si ha la massima velocità, mentre chiamandola uno scatto sì e uno no, tramite un'operazione di EOR (or esclusivo o complemento) su un registro aggiuntivo posto in $C07E, si ha la minima velocità.

La terza parte della routine è quella che si occupa del movimento degli sprite e può essere ulteriormente divisa in due sotto-parti: nella prima viene compiuto il movimento a raggiera, mentre nella seconda quello circolare, ottenuto sommando, alle due coordinate X e Y di ogni sprite, le due corrispondenti coordinate di una circonferenza tangente il bordo verticale sinistro e quello orizzontale posto in alto, sotto la cornice del video (valori in $C400 - $C5FF).

La peculiarità del programma proposto risiede nel fatto che qualsiasi utente, con un minimo(!) di nozioni geometriche e informatiche, può essere in grado di modificare a suo piacimento la traiettoria del moto.

Coloro che abbiano una qualsiasi estensione ai comandi basic (routine di Toma, Simon's Basic, GWbasic simulator,...) che permetta loro di tracciare facilmente punti in pagina grafica, provino tramite un ciclo FOR...NEXT a visualizzare le coppie X-Y di valori contenuti nelle tabelle e si accorgeranno, ad esempio, che i valori situati da $C400 a $C5FF una volta "graficati", danno origine ad una circonferenza perfetta; oppure che quelli posti da $C600 a $CBFF rappresentano una serie di raggi inclinati in vario modo con un fulcro al centro dello schermo.

Inutile dire che, cambiando i valori posti nelle tabelle, magari ricavandoli direttamente da una formula matematica (e senza scriverli uno per uno), è possibile ottenere moti sinusoidali, romboidali, a spirale e come più vi aggrada. Potreste per esempio riprodurre il tipico moto del punto luminoso di un encefalogramma!

Inoltre è stata facilitata la creazione di un moto ellittico in quanto, esaminando bene il disassemblato, si nota che sono stati impiegati due registri (locazione $C051 e $C056, registri $C05E e $C05F) per scandagliare separatamente le coordinate X e Y della circonferenza.

Ciò significa che, se inseriamo un diverso valore di partenza nel registro $C05F, la circonferenza verrà distorta assumendo le fattezze di un'ellisse.

Sfida

E ora concludiamo con la classica sfida: provate a realizzare una routine analoga, ma che sia in grado di visualizzare più di 8 sprite.

Non dovrete far altro che inserire un altro scatto raster a metà schermo, in corrispondenza del fulcro della raggiera, e ripetere le operazioni fatte per la metà superiore dello schermo. Ma attenzione: il fulcro si muove! Ciò significa che, del C/64, continueremo ad occuparci anche nei prossimi numeri? Forse sì, dipende dalle lettere di incoraggiamento che riceveremo.

Chi lo desidera, comunque, può richiedere il dischetto (e **non** il nastro), contenente i (lunghi) programmi qui pubblicati, inviando la modica somma di ventimila lire, comprese le spese di imballaggio e spedizione.

Listato n.1

```
100 rem ********************************
106 rem *         listato 1            *
120 rem * alloca i codici compattati   *
140 rem * degli sprites...             *
160 rem ********************************
170 :
180 poke53265,11: rem video off (+ veloce!)
200 fori=0to947:reada:poke2048+i,a:k=k+a:next
205 poke53265,27:ifk<>100670thenprint"err!":stop
210 printchr$(147)"tutto ok."
220 print:print"ora fai: poke44,8:run"
230 print:print"infine un bel reset!"
240 print:print"poi scrivi il listato basic n' 2"
250 end
1000 data 000,011,008,000,250,158,050
1010 data 048,054,049,000,000,000,162
1020 data 255,120,154,216,032,163,253
1030 data 032,080,253,032,021,253,032
1040 data 091,255,088,032,083,228,032
1050 data 191,227,032,034,228,169,000
1060 data 141,032,208,141,033,208,141
1070 data 134,002,032,068,229,162,071
1080 data 169,015,157,000,216,189,075
1090 data 008,157,224,003,202,016,242
1100 data 076,224,003,000,000,120,169
1110 data 052,133,001,162,000,189,147
1120 data 008,157,003,008,232,208,247
1130 data 238,233,003,238,236,003,173
1140 data 236,003,201,255,208,234,076
1150 data 011,008,019,016,018,009,020
1160 data 005,019,032,003,015,013,016
1170 data 001,020,020,001,020,009,032
1180 data 004,001,032,006,046,002,018
1190 data 049,032,010,018,032,019,015
1200 data 006,020,023,001,018,005,033
1210 data 075,067,000,040,002,002,041
1220 data 000,160,000,120,044,191,227
1230 data 234,234,234,185,077,010,153
1240 data 203,000,200,208,247,076,000
1250 data 001,012,152,047,113,000,127
1260 data 097,032,053,057,095,025,100
1270 data 155,199,230,171,028,010,249
1280 data 202,064,233,113,190,048,145
1290 data 140,068,099,035,038,136,124
1300 data 189,051,157,053,095,232,216
1310 data 136,249,002,071,060,111,190
1320 data 054,013,146,132,249,066,007
1330 data 047,227,075,157,008,249,042
1340 data 144,190,220,089,224,075,157
1350 data 010,102,107,241,005,011,190
1360 data 200,065,086,190,048,017,242
1370 data 117,002,238,047,112,052,248
1380 data 218,012,056,175,190,222,065
1390 data 018,045,098,190,048,161,074
1400 data 190,194,200,128,008,229,087
1410 data 073,042,095,216,032,002,034
1420 data 229,151,058,107,150,175,012
1430 data 152,248,010,135,220,190,232
1440 data 129,192,248,154,012,247,151
1450 data 038,125,241,085,162,103,232
1460 data 248,011,156,043,239,043,083
1470 data 233,203,028,107,125,157,048
1480 data 251,171,002,234,128,009,192
1490 data 032,176,005,008,214,095,144
1500 data 012,244,021,015,023,057,245
1510 data 226,042,006,086,005,117,004
1520 data 224,193,028,014,016,060,176
1530 data 216,005,227,020,002,176,005
1540 data 160,014,220,170,100,051,160
1550 data 002,170,021,205,234,011,156
1560 data 167,016,007,100,036,224,193
1570 data 044,046,048,167,064,128,008
1580 data 032,008,227,178,050,092,225
1590 data 252,011,039,003,064,041,006
1600 data 043,193,015,011,198,249,002
1610 data 135,139,003,230,074,231,041
1620 data 156,161,138,001,049,177,011
1630 data 011,192,041,016,032,224,248
1640 data 224,030,192,002,207,106,102
1650 data 170,253,139,158,014,178,000
1660 data 002,112,020,192,184,064,000
1670 data 091,064,002,003,128,000,174
1680 data 053,043,157,175,118,132,097
1690 data 004,130,204,085,207,023,060
1700 data 011,146,006,142,014,190,156
1710 data 075,158,175,121,024,011,010
1720 data 215,060,095,243,056,006,040
1730 data 254,085,031,239,143,012,188
1740 data 185,246,209,192,008,027,112
1750 data 000,000,015,200,200,200,169
1760 data 000,133,167,166,251,048,013
1770 data 006,250,042,038,167,198,251
1780 data 136,208,242,170,024,096,072
1790 data 161,255,133,250,162,007,134
1800 data 251,228,255,166,254,208,002
1810 data 198,255,198,254,104,144,221
1820 data 224,231,208,217,134,001,032
1830 data 191,227,133,212,076,013,008
1840 data 133,004,170,180,233,032,086
1850 data 007,166,004,117,203,072,165
1860 data 167,117,216,168,208,014,165
1870 data 002,208,010,104,201,001,024
1880 data 208,005,169,016,208,222,104
1890 data 101,252,170,152,101,253,133
1900 data 072,164,168,240,032,138,056
1910 data 229,168,176,003,198,072,056
1920 data 133,071,165,252,229,168,176
1930 data 002,198,253,133,252,177,071
1940 data 136,145,252,208,249,196,169
1950 data 240,140,177,071,198,253,198
1960 data 072,198,169,016,237,207,011
1970 data 008,199,007,158,050,048,054
1980 data 049,220,013,162,000,189,064
1990 data 008,157,000,192,232,208,247
2000 data 238,222,238,020,008,173,017
2010 data 008,201,015,208,234,096,097
2020 data 023,192,174,003,065,103,089
2030 data 058,219,249,029,052,201,057
2040 data 204,034,156,047,119,002,012
2050 data 032,065,130,255,127,120,000
2060 data 016,032,064,128,000,000,000
2070 data 000,000,000,001,002,000,000
2080 data 000,000,000,001,003,007,008
2090 data 003,004,005,006,007,007,008
2100 data 009,004,004,005,006,007,009
2110 data 010,008,008,135,000,192,014
2120 data 127,009,185,077,009,153,032
2130 data 007,200,208,247,032,085,007
2140 data 240,070,032,085,007,208,048
2150 data 032,084,007,105,002,201,004
2160 data 144,039,208,007,032,085,007
2170 data 105,004,208,030,032,083,007
2180 data 105,006,201,013,208,021,200
2190 data 032,083,007,105,013,201,013
2200 data 208,007,160,008,032,086,007
2210 data 105,029,234,234,234,234,133
2220 data 168,166,254,165,255,032,184
2230 data 007,165,072,133,255,165,071
2240 data 133,254,032,085,007,133,002
2250 data 240,020,010,044,169,003,133
2260 data 168,032,083,007,166,002,208
2270 data 002,105,008,032,143,007,240
2280 data 154,032,085,007,240,233,032
2290 data 084,007,105,004,201,006,144
2300 data 226,208,007,032,084,007,105
2310 data 006,208,217,160,005,032,086
2320 data 007,105,010,201,021,240,013
2330 data 201,022,208,202,160,001,032
```

```
2340 data 086,007,105,045,208,193,160
2350 data 006,208,245
```

Listato n. 2

```
100 rem ********************************
105 rem *                              *
106 rem *         listato 2            *
110 rem *                              *
120 rem * salva il file: "mobs bolle" *
130 rem *                              *
140 rem * su disco, in modo da poterlo*
150 rem *                              *
160 rem * allocare in $0800 -> $0e80  *
170 rem *                              *
180 rem * con un banale load"nome",8,1*
190 rem *                              *
195 rem ********************************
196 :
200 printchr$(147)"inserisci disco con almeno 7 blocchi"
205 print:print"liberi e premi un tasto quando sei ok."
210 gosub999
215 open15,8,15,"i0":gosub888
220 ifathenclose15:print"disk error!":stop
225 print:print"saving...":print
230 open8,8,8,"mobs bolle,p,w":print#8,chr$(0);chr$(8);
240 fori=49152to50815:rem $c000 -> $c67f
260 f=peek(i):print#8,chr$(f);:next
270 close8:print#15,"i0":gosub888:close15
280 end
888 input#15,a,a$,b,c:printa,a$,b,c:return
999 poke198,0:wait198,1:poke198,0:return
```

Listato n.3

```
100 rem ********************************
106 rem *         listato 3            *
110 rem *                              *
120 rem * alloca i codici compattati   *
130 rem *                              *
140 rem * delle tabelle...             *
150 rem *                              *
160 rem ********************************
170 :
180 poke53265,11: rem video off (+ veloce!)
200 fori=0to2001:reada:poke2048+i,a:k=k+a:next
205 poke53265,27:ifk<>202238thenprint"err!":stop
210 printchr$(147)"tutto ok."
220 print:print"ora fai: poke44,8:run"
230 print:print"infine un bel reset!"
240 print:print"poi scrivi il listato basic n' 4"
250 end
1000 data 000,014,008,198,007,158,050
1010 data 048,054,052,084,083,066,000
1020 data 000,000,160,160,185,029,008
1030 data 153,246,000,136,208,247,076
1040 data 008,001,173,198,248,230,248
1050 data 208,002,230,249,230,001,141
1060 data 032,208,198,001,096,120,169
1070 data 052,133,001,160,000,206,023
1080 data 001,206,026,001,185,206,015
1090 data 153,000,000,136,208,247,173
1100 data 023,001,201,007,208,234,162
1110 data 001,032,247,000,201,254,240
1120 data 038,201,252,240,044,201,253
1130 data 240,044,201,249,240,048,141
1140 data 001,008,238,059,001,208,003
1150 data 238,060,001,202,208,242,165
1160 data 249,208,217,169,055,133,001
1170 data 088,076,013,008,032,247,000
1180 data 170,032,247,000,076,058,001
1190 data 162,003,208,246,032,247,000
1200 data 170,169,000,240,208,232,208
1210 data 249,011,008,199,007,158,050
1220 data 048,054,049,253,003,162,000
1230 data 189,064,008,157,000,196,232
1240 data 208,247,238,017,008,238,020
1250 data 008,173,017,008,201,017,208
1260 data 234,096,253,026,047,048,049
1270 data 050,051,052,053,054,055,057
1280 data 058,059,060,061,062,063,064
1290 data 065,067,067,069,069,071,071
1300 data 072,073,074,075,076,077,078
1310 data 079,080,080,081,082,083,083
1320 data 084,085,085,086,087,087,088
1330 data 088,089,089,090,090,254,004
1340 data 091,252,092,254,015,093,252
1350 data 092,252,091,252,090,089,089
1360 data 088,088,087,087,086,085,085
1370 data 084,083,082,082,081,080,079
1380 data 078,078,077,076,075,074,073
1390 data 072,071,070,069,068,067,066
1400 data 065,064,063,062,061,060,059
1410 data 058,056,055,054,053,052,051
1420 data 050,048,047,046,045,044,043
1430 data 042,041,039,038,037,036,035
1440 data 034,033,032,031,029,028,027
1450 data 026,025,024,023,022,021,020
1460 data 019,018,017,017,016,015,014
1470 data 013,012,012,011,010,009,009
1480 data 008,007,007,006,006,005,005
1490 data 004,004,003,003,252,002,254
1500 data 004,001,253,014,254,004,001
1510 data 252,002,003,003,252,004,005
1520 data 006,006,007,007,008,009,009
1530 data 010,011,011,012,013,014,015
1540 data 015,016,017,018,019,020,021
1550 data 022,023,024,025,026,027,028
1560 data 029,030,031,032,033,035,036
1570 data 037,038,039,040,041,043,044
1580 data 045,046,254,008,087,254,004
1590 data 086,252,085,084,084,252,083
1600 data 082,081,081,080,080,079,079
1610 data 078,077,077,076,075,074,073
1620 data 073,072,071,070,069,068,068
1630 data 067,066,065,064,063,062,061
1640 data 060,059,058,057,056,055,054
1650 data 053,052,051,050,048,047,046
1660 data 045,044,043,042,041,040,039
1670 data 038,037,036,035,034,033,032
1680 data 031,030,029,028,026,026,025
1690 data 024,023,022,021,020,019,018
1700 data 017,016,015,015,014,013,012
1710 data 012,011,010,009,009,008,008
1720 data 007,006,006,005,005,004,004
1730 data 252,003,252,002,254,004,001
1740 data 253,014,254,005,001,252,002
1750 data 003,003,004,004,005,005,006
1760 data 006,007,007,008,009,009,010
1770 data 011,011,012,013,014,014,015
1780 data 016,017,018,019,020,020,022
1790 data 022,023,024,025,026,027,028
1800 data 029,030,031,032,033,035,036
1810 data 037,038,039,040,041,042,043
1820 data 044,045,046,047,048,049,050
1830 data 051,052,053,054,055,056,057
1840 data 058,059,060,061,062,063,064
1850 data 065,066,067,068,069,070,070
1860 data 071,072,073,074,074,075,076
1870 data 077,077,078,079,079,080,080
1880 data 081,082,082,083,083,252,084
1890 data 252,085,254,004,086,254,007
1900 data 087,254,005,137,138,138,139
1910 data 139,140,141,141,142,143,144
1920 data 146,147,148,150,151,154,155
1930 data 157,158,161,163,164,166,168
1940 data 172,174,176,178,181,183,186
1950 data 189,192,195,197,200,202,207
```

```
1960 data 209,212,215,218,221,225,228
1970 data 231,234,237,240,245,248,251
1980 data 255,002,006,010,013,016,019
1990 data 254,005,137,252,136,135,134
2000 data 134,133,132,131,130,129,128
2010 data 126,125,123,122,119,118,116
2020 data 114,112,110,108,106,104,101
2030 data 098,096,094,091,089,086,083
2040 data 080,077,075,072,069,065,063
2050 data 060,057,054,051,047,043,040
2060 data 037,034,030,027,024,020,017
2070 data 013,010,006,003,000,254,069
2080 data 137,252,136,135,134,134,133
2090 data 132,131,130,129,128,126,125
2100 data 123,122,119,118,116,114,112
2110 data 110,108,106,104,101,098,096
2120 data 094,091,089,086,083,080,077
2130 data 075,072,069,065,063,060,057
2140 data 054,051,047,043,040,037,034
2150 data 030,027,024,020,017,013,010
2160 data 006,003,000,254,005,137,138
2170 data 138,139,139,140,141,141,142
2180 data 143,144,146,147,148,150,151
2190 data 154,155,157,158,161,163,164
2200 data 166,168,172,174,176,178,181
2210 data 183,186,189,192,195,197,200
2220 data 202,207,209,212,215,218,221
2230 data 225,228,231,234,237,240,245
2240 data 248,251,255,002,006,010,013
2250 data 016,019,254,069,137,138,138
2260 data 139,139,140,141,141,142,143
2270 data 144,146,147,148,150,151,154
2280 data 155,157,158,161,163,164,166
2290 data 168,172,174,176,178,181,183
2300 data 186,189,192,195,197,200,202
2310 data 207,209,212,215,218,221,225
2320 data 228,231,234,237,240,245,248
2330 data 251,255,002,006,010,013,016
2340 data 019,254,005,137,138,138,139
2350 data 139,140,141,141,142,143,144
2360 data 146,147,148,150,151,154,155
2370 data 157,158,161,163,164,166,168
2380 data 172,174,176,178,181,183,186
2390 data 189,192,195,197,200,202,207
2400 data 209,212,215,218,221,225,228
2410 data 231,234,237,240,245,248,251
2420 data 255,002,006,010,013,016,019
2430 data 254,005,137,252,136,135,134
2440 data 134,133,132,131,130,129,128
2450 data 126,125,123,122,119,118,116
2460 data 114,112,110,108,106,104,101
2470 data 098,096,094,091,089,086,083
2480 data 080,077,075,072,069,065,063
2490 data 060,057,054,051,047,043,040
2500 data 037,034,030,027,024,020,017
2510 data 013,010,006,003,000,254,010
2520 data 137,254,004,138,254,004,139
2530 data 140,140,254,004,141,142,142
2540 data 143,143,144,144,146,146,147
2550 data 147,148,148,150,150,151,151
2560 data 154,154,155,155,157,157,158
2570 data 158,161,161,163,163,164,164
2580 data 166,166,168,168,172,172,174
2590 data 174,176,176,254,010,137,254
2600 data 006,136,135,135,254,004,134
2610 data 133,133,132,132,131,131,130
2620 data 130,129,129,128,128,126,126
2630 data 125,125,123,123,122,122,119
2640 data 119,118,118,116,116,114,114
2650 data 112,112,110,110,108,108,106
2660 data 106,104,104,101,101,098,098
2670 data 254,005,137,138,138,139,139
2680 data 140,141,141,142,143,144,146
2690 data 147,148,150,151,154,155,157
2700 data 158,161,163,164,166,168,172
2710 data 174,176,178,181,183,186,189
2720 data 192,195,197,200,202,207,209
2730 data 212,215,218,221,225,228,231
2740 data 234,237,240,245,248,251,255
2750 data 002,006,010,013,016,019,254
2760 data 069,090,252,089,252,088,087
2770 data 086,086,085,084,084,083,082
2780 data 081,080,079,078,077,076,074
2790 data 072,071,070,069,067,066,064
2800 data 063,061,060,058,056,054,052
2810 data 050,049,047,045,044,042,040
2820 data 037,035,033,031,029,027,025
2830 data 023,021,018,016,014,012,010
2840 data 008,006,004,254,005,090,252
2850 data 091,092,092,093,093,094,094
2860 data 095,096,097,097,098,099,100
2870 data 101,102,103,105,106,108,109
2880 data 111,112,113,115,116,118,119
2890 data 121,122,124,127,128,130,132
2900 data 134,135,137,139,141,144,146
2910 data 147,149,151,153,155,157,159
2920 data 162,164,166,168,171,173,175
2930 data 177,254,005,090,252,091,092
2940 data 092,093,093,094,094,095,096
2950 data 097,097,098,099,100,101,102
2960 data 103,105,106,108,109,111,112
2970 data 113,115,116,118,119,121,122
2980 data 124,127,128,130,132,134,135
2990 data 137,139,141,144,146,147,149
3000 data 151,153,155,157,159,162,164
3010 data 166,168,171,173,175,177,254
3020 data 005,090,252,089,252,088,087
3030 data 086,086,085,084,084,083,082
3040 data 081,080,079,078,077,076,074
3050 data 072,071,070,069,067,066,064
3060 data 063,061,060,058,056,054,052
3070 data 050,049,047,045,044,042,040
3080 data 037,035,033,031,029,027,025
3090 data 023,021,018,016,014,012,010
3100 data 008,006,004,254,005,090,252
3110 data 089,252,088,087,086,086,085
3120 data 084,084,083,082,081,080,079
3130 data 078,077,076,074,072,071,070
3140 data 069,067,066,064,063,061,060
3150 data 058,056,054,052,050,049,047
3160 data 045,044,042,040,037,035,033
3170 data 031,029,027,025,023,021,018
3180 data 016,014,012,010,008,006,004
3190 data 254,005,090,252,091,092,092
3200 data 093,093,094,094,095,096,097
3210 data 097,098,099,100,101,102,103
3220 data 105,106,108,109,111,112,113
3230 data 115,116,118,119,121,122,124
3240 data 127,128,130,132,134,135,137
3250 data 139,141,144,146,147,149,151
3260 data 153,155,157,159,162,164,166
3270 data 168,171,173,175,177,254,074
3280 data 090,254,006,089,254,006,088
3290 data 087,087,254,004,086,085,085
3300 data 254,004,084,083,083,082,082
3310 data 081,081,080,080,079,079,078
3320 data 078,077,077,076,076,074,074
3330 data 072,072,071,071,070,070,069
3340 data 069,067,067,066,066,254,005
3350 data 090,252,091,092,092,093,093
3360 data 094,094,095,096,097,097,098
3370 data 099,100,101,102,103,105,106
3380 data 108,109,111,112,113,115,116
3390 data 118,119,121,122,124,127,128
3400 data 130,132,134,135,137,139,141
3410 data 144,146,147,149,151,153,155
3420 data 157,159,162,164,166,168,171
3430 data 173,175,177,254,005,090,252
3440 data 091,092,092,093,093,094,094
3450 data 095,096,097,097,098,099,100
```

```
3460 data 101,102,103,105,106,108,109
3470 data 111,112,113,115,116,118,119
3480 data 121,122,124,127,128,130,132
3490 data 134,135,137,139,141,144,146
3500 data 147,149,151,153,155,157,159
3510 data 162,164,166,168,171,173,175
3520 data 177,254,010,090,254,006,089
3530 data 254,006,088,087,087,254,004
3540 data 086,085,085,254,004,084,083
3550 data 083,082,082,081,081,080,080
3560 data 079,079,078,078,077,077,076
3570 data 076,074,074,072,072,071,071
3580 data 070,070,069,069,067,067,066
3590 data 066,254,008,057,254,007,056
3600 data 254,005,055,254,004,054,254
3610 data 004,053,254,004,052,254,004
3620 data 051,254,004,050,254,004,049
3630 data 254,004,048,047,046,045,044
3640 data 043,042,041,040,039,038,037
3650 data 036,035,034,033,032,254,008
3660 data 057,254,007,056,254,005,055
3670 data 254,004,054,254,004,053,254
3680 data 004,052,254,004,051,254,004
3690 data 050,254,004,049,254,004,048
3700 data 047,046,045,044,043,042,041
3710 data 040,039,038,037,036,035,034
3720 data 033,032,254,008,057,254,007
3730 data 056,254,005,055,254,004,054
3740 data 254,004,053,254,004,052,254
3750 data 004,051,254,004,050,254,004
3760 data 049,254,004,048,047,046,045
3770 data 044,043,042,041,040,039,038
3780 data 037,036,035,034,033,032,254
3790 data 008,057,254,007,056,254,005
3800 data 055,254,004,054,254,004,053
3810 data 254,004,052,254,004,051,254
3820 data 004,050,254,004,049,254,004
3830 data 048,047,046,045,044,043,042
3840 data 041,040,039,038,037,036,035
3850 data 034,033,032,000,000,000,000
```

Listato n.4

```
100 rem ********************************
105 rem *                              *
106 rem *          listato 4           *
110 rem *                              *
120 rem * alloca i codici macchina     *
130 rem *                              *
140 rem * per la routine e la tabella *
150 rem *                              *
160 rem * per le x > di 255...         *
170 rem *                              *
180 rem ********************************
190 :
200 fori=0to160:poke49152+i,0:next
205 p=1:n=89:lo=49152:gosub900
210 printchr$(147)"wedge ok."
220 p=2:n=13:lo=49264:gosub900
230 print"start ok."
240 p=3:n=614:lo=49408:gosub900
250 print"move ok."
260 print:print"clearing & writing control map...";
270 fori=52480to53247:pokei,0:next
310 for y = 1 to 5: read lo: for i = lo to lo+5: poke i,62:
next i,y: print"ok!"
320 print:print"scegli la velocita' dell'effetto:"
330 print"1 - lenta": print"2 - media (default)": print"3
elevata"
340 print:ve=0:inputve
350 ifve<1orve>3then320
360                                                       if
ve=3thenlo=49272:pokelo,32:pokelo+1,0:pokelo+2,193:goto400
370 if ve = 2 then400
380 lo = 49270: poke lo,128: poke lo+1,192: lo=49280: for
i=0 to 13: reada:pokelo+i,a:next
400 ef=0:print:input"quale effetto (1=bolle 2=stelle)";ef
410 ifef<1oref>2then400
420 ifef=1then490
430 poke 49189,142: poke 49193,11: lo=52224: fori=0to255:
poke lo+i,13: next:poke832,1
490 print:print"ora resetta e carica il listato 5..."
899 end
900 k=0:fori=0ton:reada:pokelo+i,a:k=k+a:next
910 readc:ifk<>cthenprint"errore parte ";p:stop
920 return
1000 rem data wedge
1001 :
1002 data 120,169,192,162,112,141,021
1010 data 003,142,020,003,173,017,208
1020 data 041,127,141,017,208,162,001
1030 data 142,026,208,202,142,014,220
1040 data 142,018,208,088,169,255,141
1050 data 021,208,141,028,208,169,006
1060 data 157,039,208,232,224,008,208
1070 data 248,169,014,141,038,208,169
1080 data 003,141,037,208,160,008,162
1090 data 000,138,157,096,192,024,105
1100 data 008,232,136,208,246,140,032
1110 data 208,140,033,208,140,094,192
1120 data 169,000,141,095,192,096,11140
1121 :
1122 rem data start
1123 :
1130 data 169,001,141,025,208,032,000
1140 data 193,234,234,234,076,049,234,1830
1141 :
1142 rem data move
1143 :
2000 data 174,096,192,189,000,198,141
2010 data 000,208,189,000,201,141,001
2020 data 208,189,000,204,141,248,007
2030 data 189,000,205,208,008,169,000
2040 data 141,104,192,076,039,193,169
2050 data 001,141,104,192,238,096,192
2060 data 234,174,097,192,189,000,199
2070 data 141,002,208,189,000,202,141
2080 data 003,208,189,000,204,141,249
2090 data 007,189,000,206,208,008,169
2100 data 000,141,105,192,076,082,193
2110 data 169,002,141,105,192,238,097
2120 data 192,234,174,098,192,189,000
2130 data 198,141,004,208,189,000,201
2140 data 141,005,208,189,000,204,141
2150 data 250,007,189,000,205,208,008
2160 data 169,000,141,106,192,076,125
2170 data 193,169,004,141,106,192,238
2180 data 098,192,238,098,192,234,174
2190 data 099,192,189,000,199,141,006
2200 data 208,189,000,202,141,007,208
2210 data 189,000,204,141,251,007,189
2220 data 000,206,208,008,169,000,141
2230 data 107,192,076,171,193,169,008
2240 data 141,107,192,238,099,192,238
2250 data 099,192,234,174,100,192,189
2260 data 000,200,141,008,208,189,000
2270 data 203,141,009,208,189,000,204
2280 data 141,252,007,189,000,207,208
2290 data 008,169,000,141,108,192,076
2300 data 217,193,169,016,141,108,192
2310 data 238,100,192,234,174,101,192
2320 data 189,000,200,141,010,208,189
2330 data 000,203,141,011,208,189,000
2340 data 204,141,253,007,189,000,207
2350 data 208,008,169,000,141,109,192
2360 data 076,004,194,169,032,141,109
2370 data 192,238,101,192,238,101,192
2380 data 234,174,102,192,189,000,200
```

```
2390 data 141,012,208,189,000,203,141
2400 data 013,208,189,000,204,141,254
2410 data 007,189,000,207,208,008,169
2420 data 000,141,110,192,076,050,194
2430 data 169,064,141,110,192,238,102
2440 data 192,234,174,103,192,189,000
2450 data 200,141,014,208,189,000,203
2460 data 141,015,208,189,000,204,141
2470 data 255,007,189,000,207,208,008
2480 data 169,000,141,111,192,076,093
2490 data 194,169,128,141,111,192,238
2500 data 103,192,238,103,192,234,234
2510 data 234,174,095,192,173,001,208
2520 data 024,125,000,197,141,001,208
2530 data 173,003,208,024,125,000,197
2540 data 141,003,208,173,005,208,024
2550 data 125,000,197,141,005,208,173
2560 data 007,208,024,125,000,197,141
2570 data 007,208,173,009,208,024,125
2580 data 000,197,141,009,208,173,011
2590 data 208,024,125,000,197,141,011
2600 data 208,173,013,208,024,125,000
2610 data 197,141,013,208,173,015,208
2620 data 024,125,000,197,141,015,208
2630 data 238,095,192,234,174,094,192
2640 data 173,000,208,024,125,000,196
2650 data 141,000,208,144,005,169,001
2660 data 141,104,192,173,002,208,024
2670 data 125,000,196,141,002,208,144
2680 data 005,169,002,141,105,192,173
2690 data 004,208,024,125,000,196,141
2700 data 004,208,144,005,169,004,141
2710 data 106,192,173,006,208,024,125
2720 data 000,196,141,006,208,144,005
2730 data 169,008,141,107,192,173,008
2740 data 208,024,125,000,196,141,008
2750 data 208,144,005,169,016,141,108
2760 data 192,173,010,208,024,125,000
2770 data 196,141,010,208,144,005,169
2780 data 032,141,109,192,173,012,208
2790 data 024,125,000,196,141,012,208
2800 data 144,005,169,064,141,110,192
2810 data 173,014,208,024,125,000,196
2820 data 141,014,208,144,005,169,128
2830 data 141,111,192,238,094,192,173
2840 data 104,192,013,105,192,013,106
2850 data 192,013,107,192,013,108,192
2860 data 013,109,192,013,110,192,013
2870 data 111,192,141,016,208,096,77630
3000 :
3001 rem data tabella x > 255
3002 :
3003 data52538,52794,52922,52986,53242
4000 :
4001 rem data velocita' lenta
4002 :
4003 data 173,126,192,073,255,141,126
4004 data 192,208,003,032,000,193,096
```

Listato N. 5

```
100 rem ******************************
105 rem *                            *
106 rem *        listato 5           *
110 rem *                            *
120 rem * salva il file: "bolle pazze"*
130 rem *                            *
140 rem * su disco, in modo da poterlo*
150 rem *                            *
160 rem * allocare in $c000 -> $d000  *
170 rem *                            *
180 rem * con un banale load"nome",8,1*
190 rem *                            *
195 rem ******************************
196 :
200 printchr$(147)"inserisci disco con almeno 17 blocchi"
205 print:print"liberi e premi un tasto quando sei ok."
210 gosub999
215 open15,8,15,"i0":gosub888
220 ifathenclose15:print"disk error!":stop
225 print:print"saving...":print
230 open 8, 8, 8, "bolle pazze, p, w" : print#8, chr$(0);
chr$(192);
240 fori=49152to53247:rem $c000 -> $d000
260 f=peek(i):print#8,chr$(f);:next
270 close8:print#15,"i0":gosub888:close15
280 end
888 input#15,a,a$,b,c:printa,a$,b,c:return
999 poke198,0:wait198,1:poke198,0:return
```

Disassemblato commentato

```
;-------------------------------
;- bolle pazze! / disassemblato -
;-                              -
;-   disassemblato commentato   -
;-                              -
;-  f.brl jr software dicembre 91-
;-                              -
;-------------------------------
        vic      = $d000
        hirast   = $d011
        raster   = $d012
        bordo    = $d020
        sfondo   = $d021
        irqext   = $ea31
        nmob     = $07f8
;-------------------------------
        xmap     = $c060
        xincr    = $c05e
        yincr    = $c05f
        x02      = $c600
        x13      = $c700
        x4567    = $c800
        y02      = $c900
        y13      = $ca00
        y4567    = $cb00
        vector   = $cc00
        xrot     = $c400
        yrot     = $c500
        x02mag   = $cd00
        x13mag   = $ce00
        x47mag   = $cf00
        outreg   = $c068
;-------------------------------
        *= $c000
        sei             ;disabilita l'irq
begin   lda #$c0        ;lo punta a $c070
        ldx #$70        ;dove inizia la routine
        sta $0315       ;
        stx $0314       ;
        lda hirast      ;pone lo 'scatto' raster
        and #$7f        ;per l'irq nella parte
        sta hirast      ;alta del video
        ldx #$01        ;abilita gli irq causati
        stx $d01a       ;da uno scatto raster
        dex             ;pone a 0 il timer
        stx $dc0e       ;
        stx raster      ;scatto alla riga 0
        cli             ;riabilita l'irq
        lda #$ff        ;accende tutti gli
        sta vic+21      ;sprites e il
        sta vic+28      ;modo multicolor
        lda #$06        ;colore 1 = blu
loop    sta vic+39,x    ; (per tutti gli sprites)
        inx             ;
        cpx #$08        ;
        bne loop        ;
```

```
        lda #$0e        ;colore 2 = celeste
        sta vic+38      ;
        lda #$03        ;colore 3 = blu/verde (ciano)
        sta vic+37      ;
        ldy #$08        ;riempie la mappa per
        ldx #$00        ;i contatori x con
        txa             ;valori da 0 a 56
loop2   sta xmap,x      ;multipli di 8
        clc             ;
        adc #$08        ;
        inx             ;
        dey             ;
        bne loop2       ;
        sty bordo       ;bordo nero
        sty sfondo      ;sfondo nero
        sty xincr       ;incremento x di partenza=0
        lda #$00        ;stessa cosa per
        sta yincr       ;l'incremento y
        rts             ;esce in basic
;---------------------------------
        *=$c070         ;
start   lda #$01        ;conferma la chiamata
        sta $d019       ;dell'irq
        jsr move        ;routine per il movimento
        nop             ;spazio libero per un
        nop             ;salto facoltativo
        nop             ;(cfr. articolo)
        jmp irqext      ;scansione tastiera(esce)
;---------------------------------
        *=$c100         ;
move    ldx xmap        ;legge la x
        lda x02,x       ;legge l'x-esimo valore dalla
        sta vic         ;tabella delle x e y per
        lda y02,x       ;gli sprites 0 e 2
        sta vic+1       ;e li mette nei loro registri
        lda vector,x    ;legge la 'figura' corrispondente
        sta nmob        ;al mob in quella posizione
        lda x02mag,x    ;controlla se la x e' > 255
        bne out1        ;se si' allora salta
        lda #$00        ;resetta il registro per
        sta outreg      ;x out (>) 255
        jmp next1       ;
out1    lda #$01        ;setta il registro per
        sta outreg      ;la x > 255
next1   inc xmap        ;incrementa il contatore x
        nop             ;separatore estetico!
        ldx xmap+1      ;da ora in poi si compiono
        lda x13,x       ;le medesime operazioni
        sta vic+2       ;per determinare la
        lda y13,x       ;posizione, la 'figura'
        sta vic+3       ;(per intenderci: la
        lda vector,x    ;dimensione della bolla)
        sta nmob+1      ;e l'eventuale superamento
        lda x13mag,x    ;della posizione 255
        bne out2        ;di ognuno dei 7 sprites
        lda #$00        ;rimanenti...
        sta outreg+1    ;
        jmp next2       ;
out2    lda #$02        ;
        sta outreg+1    ;
next2   inc xmap+1      ;
        nop             ;
        ldx xmap+2      ;
        lda x02,x       ;
        sta vic+4       ;
        lda y02,x       ;
        sta vic+5       ;
        lda vector,x    ;
        sta nmob+2      ;
        lda x02mag,x    ;
        bne out3        ;
        lda #$00        ;
        sta outreg+2    ;
        jmp next3       ;
out3    lda #$04        ;
        sta outreg+2    ;
next3   inc xmap+2      ;un doppio incremento come
        inc xmap+2      ;questo provoca un movimento
        nop             ;piu' veloce della bolla!
        ldx xmap+3      ;
        lda x13,x       ;
        sta vic+6       ;
        lda y13,x       ;
        sta vic+7       ;
        lda vector,x    ;
        sta nmob+3      ;
        lda x13mag,x    ;
        bne out4        ;
        lda #$00        ;
        sta outreg+3    ;
        jmp next4       ;
out4    lda #$08        ;
        sta outreg+3    ;
next4   inc xmap+3      ;
        inc xmap+3      ;
        nop             ;
        ldx xmap+4      ;
        lda x4567,x     ;
        sta vic+8       ;
        lda y4567,x     ;
        sta vic+9       ;
        lda vector,x    ;
        sta nmob+4      ;
        lda x47mag,x    ;
        bne out5        ;
        lda #$00        ;
        sta outreg+4    ;
        jmp next5       ;
out5    lda #$10        ;
        sta outreg+4    ;
next5   inc xmap+4      ;
        nop             ;
        ldx xmap+5      ;
        lda x4567,x     ;
        sta vic+10      ;
        lda y4567,x     ;
        sta vic+11      ;
        lda vector,x    ;
        sta nmob+5      ;
        lda x47mag,x    ;
        bne out6        ;
        lda #$00        ;
        sta outreg+5    ;
        jmp next6       ;
out6    lda #$20        ;
        sta outreg+5    ;
next6   inc xmap+5      ;
        inc xmap+5      ;
        nop             ;
        ldx xmap+6      ;
        lda x4567,x     ;
        sta vic+12      ;
        lda y4567,x     ;
        sta vic+13      ;
        lda vector,x    ;
        sta nmob+6      ;
        lda x47mag,x    ;
        bne out7        ;
        lda #$00        ;
        sta outreg+6    ;
        jmp next7       ;
out7    lda #$40        ;
        sta outreg+6    ;
next7   inc xmap+6      ;
        nop             ;
        ldx xmap+7      ;
        lda x4567,x     ;
        sta vic+14      ;
        lda y4567,x     ;
        sta vic+15      ;
```

```
        lda vector,x ;
        sta nmob+7   ;
        lda x47mag,x ;
        bne out8     ;
        lda #$00     ;
        sta outreg+7 ;
        jmp next8    ;
out8    lda #$80     ;
        sta outreg+7 ;
next8   inc xmap+7   ;
        inc xmap+7   ;
        nop          ;
        nop          ;
        nop          ;
rotate  ldx yincr    ;legge il valore y di partenza
        lda vic+1    ;aggiunge alla coordinata y
        clc          ;del mob 0 ora letta
        adc yrot,x   ;il valore x-esimo preso
        sta vic+1    ;dalla tabella.
        lda vic+3    ;si procede nel medesimo
        clc          ;modo per sommare questo
        adc yrot,x   ;valore x-esimo alla
        sta vic+3    ;coordinata y di tutti
        lda vic+5    ;gli sprites.
        clc          ;
        adc yrot,x   ;
        sta vic+5    ;
        lda vic+7    ;
        clc          ;
        adc yrot,x   ;
        sta vic+7    ;
        lda vic+9    ;
        clc          ;
        adc yrot,x   ;
        sta vic+9    ;
        lda vic+11   ;
        clc          ;
        adc yrot,x   ;
        sta vic+11   ;
        lda vic+13   ;
        clc          ;
        adc yrot,x   ;
        sta vic+13   ;
        lda vic+15   ;
        clc          ;
        adc yrot,x   ;
        sta vic+15   ;
        inc yincr    ;viene incrementato il valore
        nop          ;y di partenza.
        ldx xincr    ;si legge quello della x
        lda vic      ;aggiunge alla coordinata x
        clc          ;del mob 0 ora letta
        adc xrot,x   ;il valore x-esimo preso
        sta vic      ;dalla tabella.
        bcc min1     ;se il carry=0 -> salta
        lda #$01     ;setta il registro per le
        sta outreg   ;x > 255
min1    lda vic+2    ;continua nel medesimo
        clc          ;modo per gli sprites da 1 a 7
        adc xrot,x   ;settando i corrispondenti
        sta vic+2    ;8 registri qualora le x
        bcc min2     ;fossero > 255
        lda #$02     ;con valori dati dalla
        sta outreg+1 ;formula v=2^n dove:
min2    lda vic+4    ;v= valore da inserire
        clc          ;n= numero sprite (0->7)
        adc xrot,x   ;
        sta vic+4    ;
        bcc min3     ;
        lda #$04     ;nota sul funzionamento
        sta outreg+2 ;del carry (riporto):
min3    lda vic+6    ;avendolo posto a zero
        clc          ;(come in questa riga)
        adc xrot,x   ;rimarra' tale se
        sta vic+6    ;a+b=c con c<=255
        bcc min4     ;se invece c eccedera' il
        lda #$08     ;valore 255 (ripartendo
        sta outreg+3 ;quindi da 0)il carry
min4    lda vic+8    ;verra' posto a 1.
        clc          ;
        adc xrot,x   ;
        sta vic+8    ;
        bcc min5     ;questa istruzione significa:
        lda #$10     ;'salta se il carry e' 0
        sta outreg+4 ;(branch if carry cleared)
min5    lda vic+10   ;
        clc          ;
        adc xrot,x   ;
        sta vic+10   ;
        bcc min6     ;
        lda #$20     ;
        sta outreg+5 ;
min6    lda vic+12   ;
        clc          ;
        adc xrot,x   ;
        sta vic+12   ;
        bcc min7     ;
        lda #$40     ;
        sta outreg+6 ;
min7    lda vic+14   ;
        clc          ;
        adc xrot,x   ;
        sta vic+14   ;
        bcc min8     ;
        lda #$80     ;
        sta outreg+7 ;
min8    inc xincr    ;viene incrementato il valore
        lda outreg   ;x di partenza
        ora outreg+1 ;si sommano i bit (non byte!)
        ora outreg+2 ;contenuti negli 8 registri
        ora outreg+3 ;a partire da outreg e
        ora outreg+4 ;si deposita il risultato,
        ora outreg+5 ;presente in outreg,
        ora outreg+6 ;nel registro effettivo
        ora outreg+7 ;
        sta $d010    ;per x > 255
        rts          ;fine routine!
        .end
```